
Projektthema

Qualitätssicherungsdokument

Gruppe 22: Dominik Wittenberg <dominik.wittenberg@stud.tu-darmstadt.de>
 Paul Youssef <paul.youssef@stud.tu-darmstadt.de>
 Pavel Nachname <pavel_andreevic.azanov@stud.tu-darmstadt.de>
 Robin Voigt <robin.voigt@stud.tu-darmstadt.de>
 Alessandro Noli <alessandro_fabio.noli@stud.tu-darmstadt.de>

Teamleiter: Julia Hofmann <juliahofmann92@web.de>

Auftraggeber: Marcel Schaub <schaub@mathematik.tu-darmstadt.de>
 Arbeitsgruppe Didaktik
 Fachbereich Mathematik

Abgabedatum: xx.xx.xxxx



TECHNISCHE
UNIVERSITÄT
DARMSTADT

Bachelor-Praktikum SoSe 2015
Fachbereich Informatik

Inhaltsverzeichnis

1. Einleitung	2
2. Qualitätsziele	3
2.1. Bedienbarkeit	3
2.2. Codequalität	4
2.3. Erweiterbarkeit	4
A. Anhang	5

1 Einleitung

Dieses Dokument dient zur Überprüfung der Qualitätssicherung des Projektes Entwicklung und Implementierung eines Plugins zur Ablaufsteuerung von Moodle-Tests. Moodle ist ein freies objektorientiertes Kursmanagementsystem und eine Lernplattform, welche im Jahre 2002 erschien und bis heute weiterentwickelt wird. In Moodle gibt es daher eine Trennwand zwischen dem Tutor und dem Lernenden.

Hierbei ist die Aufgabe ein Plugin zu erstellen, um den Tutor die Möglichkeit zu geben eine neue Art von Tests zu erstellen. Genauer gesagt hat das Plugin die Aufgabe, dem Prüfer die Möglichkeit zu geben Fragen zu erstellen, die ggf. weitere Unterfragen enthalten. Diese Unterfragen werden bei fehlerhafter Antwort seitens des Lernenden aufgerufen. Die Unterfragen besitzen weitere Abzweigungen und eine festgelegte Reihenfolge, die der Tutor selbst festlegen kann. Dadurch kann der Tutor auf fehlerhafte Antworten/Eingaben besser eingehen.

Das Plugin kann nach seiner Fertigstellung frei benutzt werden. Um auch Internationalen anderen Nutzern die Chance zu geben dieses Plugin zu verwenden, legen wir einen großen Wert auf eine mehrsprachige Dokumentation, die unser Plugin funktional beschreibt und dessen Bedienbarkeit.

Durch die Weiterentwicklung von Moodle werden neue Funktionen hinzugefügt. Um dem Plugin die Möglichkeit zu geben die neuen Funktionen nutzen zu können, muss das Plugin in die Ordnerstruktur von Moodle integrierbar sein. Aber auch die Weiterentwicklung seitens von anderen Entwicklern muss gewährleistet werden können.

Das Projekt wird in PHP geschrieben mit der Verwendung von der Moodle API. Als CI-Tool verwenden wir das Tool Travis, welches für Open-Source Produkte kostenlos ist. Für service Control verwenden wir Git mit einem Remotespeicher auf Github, desweiteren arbeiten wir auf einem externen Moodle server dieser nur uns zur verfügung steht. Zum Zeitmanagement verwenden wir das Online Tool Toggl, womit wir alle untereinander unsere Zeiten einsehen und überprüfen können. Die User-Stories beschreiben wir online mit Trello.

2 Qualitätsziele

Im Folgendem wird erklärt, wieso das Qualitätsziel wichtig ist und welche Maßnahmen wir getroffen haben

2.1 Bedienbarkeit

Das Plugin wird nach der Fertigstellung aktiv im Mathebereich verwendet, sowie ggf. von anderen Uni-externen Nutzern die gefallen an der Funktionalität des Moodle Plugins haben. Da die Funktionalität dieses Plugins über eine GUI bedient wird, muss diese vor allem verständlich/intuitiv gegenüber dem Nutzer sein und gut in das Moodle optisch integrierbar sein.

Dem Auftraggeber ist die Funktionalität des Plugins am wichtigsten. Da der Auftraggeber aber nur das Front-End sieht, wird das Front-End mit der Funktionalität in Absprache mit dem Arbeitgeber während mehreren Iterationen entwickelt

Daher gehört die Nutzbarkeit des Plugins zu einem wichtigen QS-Ziel, welcher im Laufe des Projekts gewährleistet werden soll. Um das QS-Ziel zu gewährleisten wollen wir eine Plugin Dokumentation schreiben. Die Plugin Dokumentation soll darauf eingehen, wie das Plugin zu benutzen ist und was man damit machen kann. Diese Dokumentation werden wir mit dem Plugin als Paket mitliefern. Zu finden ist es dann auf Github als README.me und auf der moodle.org Plugin Seite als eigenständig erzeugte Seite mit einem Moodle Account. Wir werden die

Da die Funktionalität des Plugins über das Front-End bestimmt wird, ist es wichtig, dass das Plugin für den Nutzer einfach zu verwenden ist. Im Rahmen dessen werden wir eine Nutzerstudie, mit Einarbeitungszeit in das Plugin, durchführen. Dafür werden wir Ende Januar/Anfang Februar 2016 mehrere Tutoren über das Plugin befragen. Diese sollen dann die Möglichkeit bekommen, uns nachträglich Feedback zu geben, welches wir nutzen können um die Bedienbarkeit der Nutzer zu verbessern.

Mit einer guten Dokumentation und einem Feedback von ggf. mehreren aktiven Moodle Nutzern wollen wir das Qualitätsziel der Benutzbarkeit/Bedienbarkeit gewährleisten.

2.2 Codequalität

Moodle ist ein vorhandenes System auf das wir mit unserem Plugin aufbauen. Damit wir das Plugin in Moodle gut integrieren können, müssen wir dessen Richtlinien folgen. Diese Richtlinien sind gegeben durch Styleguides, Namenskonventionen von Klassen und Methoden, sowie Ordnerstrukturen. Damit das Plugin korrekt in das Moodle System integriert wird. Moodle ist international vertreten und open-source, verwenden wir eine Dokumentation auf Englisch.

Um dies zu gewährleisten verwenden wir Code Reviews mit Checklisten. Jede User-Story muss diese Checkliste erfüllen. Es gibt immer für die jeweilige User Story einen Entwickler und einen Reviewer. Der Reviewer prüft mit der Checkliste die User Story und zeigt dem Entwickler der User Story die gefundenen Fehler. Der Entwickler muss bis zur nächsten Iteration den Fehler in seiner Implementierung der User Story behoben haben. Die Checkliste baut sich aus folgenden Teilen auf:

1. Ist der Style Guide seitens Moodle erfüllt?
2. Wurde die Namenskonvention seitens Frankenstyle erfüllt?
3. Ist das die geg. Ordnerstruktur?
4. Ist die Dokumentation auf Englisch?

Die Codequalität hilft uns und anderen Entwicklern den Code besser verstehen und erweitern zu können. Durch die Einführung von Code Reviews mit Hilfe von einer Checkliste wollen wir das Qualitätsziel von guter Codequalität erreichen.

2.3 Erweiterbarkeit

Moodle ist ein Stück Software welches täglich weiter entwickelt wird. Damit das Plugin mit Moodle zusammen wächst müssen Gegebenheiten geschaffen werden um das Plugin zu erweitern. Da Moodle Open-Source ist und das Plugin ebenfalls unter derselben Lizenz entwickelt wird, muss für andere Entwickler die Möglichkeit gegeben werden unser Plugin erweitern zu können. Darunter fällt nicht nur neue Funktionalität, sondern auch die Integration mit einem anderen Plugin.

Daher muss das Plugin Modular gemäß der Moodle Vorschriften für Sub Plugins implementiert werden. Dies bedeutet, wir greifen auf die vorgegeben Design Patterns seitens Moodle zurück, um anderen die Möglichkeit zu geben es funktional zu erweitern oder zu integrieren. Durch geeignete Code Reviews, die nach jeder Iteration stattfinden, wollen wir bis zum Projektende den Modularen Aufbau gewährleisten. Die Code Reviews umfassen die Codequalität, sowie die Möglichkeit das Plugin erweitern zu können.

A Anhang

(Am Ende des Projekts nachzureichen)

Beleg für durchgeführte Maßnahmen, bzw. falls nicht durchgeführt eine Begründung wieso die Durchführung nicht möglich oder nicht erfolgt ist.

Weitere Anforderungen sind den Unterlagen und der Vorlesung zur Projektbegleitung zu entnehmen.